

# TP 4 – Services bancaires en CORBA

## 1 Nommage

Dans le repertoire *1-Nommage*, vous trouverez un exemple d'utilisation du service de noms. Cet exemple est compilable (commande `./compile`) et exécutable (les commandes à exécuter sont déductibles des fichiers *Serveur.java* et *Client\*.java*).

À travers les questions suivantes, on vous demande de mettre au point certains services pour une banque.

## 2 Service bancaire

Cet exercice se fera dans le répertoire *2-ServiceBancaire*.

### 2.1 Interface de projection

Créez un fichier *Compte.idl*, représentant un compte bancaire. Définissez-y les services suivants qui pourront s'appliquer à un certain compte client :

```
// Déposer un certain montant sur un compte
void deposeLiquide(Flottant) ;

// Retirer d'argent du compte (mais vous n'autorisez pas le découvert !)
Boolean retireLiquide(Flottant) ;

// Afficher le montant déposé sur le compte
Flottant afficheMontant() ;

// Virer une somme du compte courant vers le compte specifié en parametre.
// Si le montant du compte est insuffisant l'operation n'est pas réalisée
// et false est retournée
Boolean virementCompteCompte(Flottant, Compte) ;
```

### 2.2 Implémentation de l'interface

Réalisez une implémentation *CompteImpl.java* pour l'interface *Compte.idl*. Créez un fichier *./compile* qui vous permet de compiler l'interface *Compte.idl* et le fichier *CompteImpl.java*.

### 2.3 Serveur

Réalisez un Serveur (*Serveur.java*) qui crée deux comptes client et les publie sur le serveur de nom. Le nom de la machine hébergeant le serveur de noms sera passé en argument du programme.

### 2.4 Client

Réalisez un Client (*Client.java*) qui :

- ajoute une somme sur les deux comptes du Serveur ;
- retire une autre somme sur chaque compte du Serveur ;

— fait un virement d'un compte a l'autre.  
Entre chaque operation vous afficherez un message qui donne le solde de chacun des comptes.

### 3 Service bancaire avec nommage

Cet exercice se fera dans le répertoire *3-ServiceBancaire*.

#### 3.1 Serveur

Construisez un deuxième Serveur (*Serveur.java*) qui crée un contexte de nommage. Celui-ci correspond à un nom d'agence bancaire. Dans cette agence bancaire, vous créez deux contextes correspondants à deux noms de clients différents. Vous créez enfin deux comptes par client qui seront associés à des objet CORBA de type *Compte*.

#### 3.2 Client

Réalisez un Client (*Client1.java*) qui utilise le nouveau *Serveur* et :

- ajoute une somme sur les 4 comptes du Serveur ;
- retire une autre somme sur chaque compte du Serveur ;
- fait un virement d'un compte vers un autre.

Entre chaque opération vous afficherez un message donnant les montants sur les quatre comptes.

#### 3.3 Encore un client

Réaliser un petit Client (*Client2.java*) qui liste les différentes agences puis liste les clients de chacune des agences trouvées. Vous utiliserez pour cela la methode *list* de la classe *NamingContext* (voir l'API java).

#### 3.4 Tests

Vérifiez vos solutions dans une configuration à plusieurs machines. Idéalement, le serveur de noms, le serveur et le client se situent sur trois machines distinctes.

## 4 Mystère

Comprendre et essayer les programmes *Client* et *Serveur* du repertoire *4-Mystere*. Répondre aux questions suivantes dans le fichier *reponse.txt* :

1. Expliquez succinctement ce que font les deux programmes.
2. Expliquez pourquoi la commande suivante renvoie un resultat valide :

```
java Client localhost 3000 rep1/rep3/rep3/rep1/rep3/rep3/rep2/et2
Résultat trouvé : Lesmemealamaison Leon-Jay
```